Amendments to the Specification:

Please replace the paragraph beginning at page 7, line 8 with the following amended paragraph:

Figures 9a and 9b ~~Figure 9~~ is a diagram illustrating a ML sparse decoder adapted to accommodate the Release-5 version of the 3GPP standard; and

Please replace the paragraph beginning at page 8, line 17 with the following amended paragraph:

Looking now at Figures 4a and 4b ~~Figure 4~~, a diagram illustrates a sparse decoder 100 implementation for the (32, 10) Reed Muller Code in accordance with one embodiment of the present invention. Given the TFCS size, the "Hypothesis Generator" 102 iterates over all the possible values of the TFCI. Usually this number will be pretty small and that is where this decoder 100 structure will yield the most benefit in terms of performance improvement and power savings described in more detail herein below. The case where the size of the TFCS is equal to 1024 (maximum possible) is only in the case of a logically split TFCI. This is the case when a DPCH channel in addition to a DSCH channel is used and the TFCI of both the channels is transmitted over the downlink DPCCH. In this scenario, the 5-bit TFCI of the DPCH is concatenated with the 5-bit TFCI of the DSCH channel and then the 10-bits are encoded using the (32,10) Reed Muller code.

Please replace the paragraph beginning at page 10, line 17 with the following amended paragraph:

The present inventors recognized the structure shown in Figures 4a and 4b ~~Figure 4~~ can be made flexible enough to readily support the (16,5) Reed Muller code in addition to the (32,10) code. The basis vectors for the (16,5) code will be used in such a scenario and the decoder will be run twice to decode both the TFCIs, one for the DPCH channel and the other for the DSCH channel. The basis vectors for this code are shown in Table 2 below.

Please replace the paragraph beginning at page 16, line 11 with the following amended paragraph:

The sparse decoder 100 can be easily modified to be used with hard-split TFCI decoding described in the Release-5 version of the 3GPP standard. The sparse decoder 200 shown in Figures 9a and 9b ~~Figure 9~~ can decode this flexible hard-split TFCI decoding in a single pass; whereas a conventional Green Machine has to run twice: one for DPCH with setting the received DSCH TFCI symbols to zero (puncturing) and a 5-bit TFCI for DSCH. In Release-5, this ratio may be different. Figure 8 illustrates the TFCI encoding procedure used in Release-5 for flexible hard split mode; and as stated herein before, can be easily implemented with the structure of the sparse decoder 200 shown in Figures 9a and 9b ~~Figure 9~~ described in greater detail herein below.